

# Quantum Key Distribution in the Classical Authenticated Key Exchange Framework

Michele Mosca<sup>1,2</sup>, Douglas Stebila<sup>3</sup>, and Berkant Ustaoglu<sup>4</sup>

<sup>1</sup> Institute for Quantum Computing and Dept. of Combinatorics & Optimization  
University of Waterloo, Waterloo, Ontario, Canada

<sup>2</sup> Perimeter Institute for Theoretical Physics, Waterloo, Ontario, Canada

[mmosca@uwaterloo.ca](mailto:mmosca@uwaterloo.ca)

<sup>3</sup> Information Security Institute, Queensland University of Technology, Brisbane, Queensland, Australia

[stebila@qut.edu.au](mailto:stebila@qut.edu.au)

<sup>4</sup> Department of Mathematics, Izmir Institute of Technology, Urla, Izmir, Turkey

[bustaoglu@uwaterloo.ca](mailto:bustaoglu@uwaterloo.ca)

June 27, 2012

## Abstract

Key establishment is a crucial primitive for building secure channels: in a multi-party setting, it allows two parties using only public authenticated communication to establish a secret session key which can be used to encrypt messages. But if the session key is compromised, the confidentiality of encrypted messages is typically compromised as well. Without quantum mechanics, key establishment can only be done under the assumption that some computational problem is hard. Since digital communication can be easily eavesdropped and recorded, it is important to consider the secrecy of information anticipating future algorithmic and computational discoveries which could break the secrecy of past keys, violating the secrecy of the confidential channel.

Quantum key distribution (QKD) can be used generate secret keys that are secure against any future algorithmic or computational improvements. QKD protocols still require authentication of classical communication, however, which is most easily achieved using computationally secure digital signature schemes. It is generally considered folklore that QKD when used with computationally secure authentication is still secure against an unbounded adversary, provided the adversary did not break the authentication during the run of the protocol.

We describe a security model for quantum key distribution based on traditional classical authenticated key exchange (AKE) security models. Using our model, we characterize the long-term security of the BB84 QKD protocol with computationally secure authentication against an eventually unbounded adversary. By basing our model on traditional AKE models, we can more readily compare the relative merits of various forms of QKD and existing classical AKE protocols. This comparison illustrates in which types of adversarial environments different quantum and classical key agreement protocols can be secure.

**Keywords:** quantum key distribution, authenticated key exchange, cryptographic protocols, security models

## 1 Introduction

Quantum key distribution (QKD) promises new security properties compared to cryptography based on computational assumptions: QKD can provide for two parties to establish a secure key using an untrusted quantum channel and a public, authenticated classical channel, and this key is secure against any adversary

who is limited solely by the laws of quantum mechanics. While some classical<sup>1</sup> cryptographic tasks can be achieved with information-theoretic security against unbounded adversaries, key establishment over a public authenticated channel is not one of them. Moreover, the practicality of such information-theoretically secure schemes is often limited, and as a result most classical cryptographic schemes rely for their security on various computational assumptions, the most widely used of which — factoring, discrete logarithms — could be efficiently solved by a large-scale quantum computer. As a result, QKD could be an important primitive for cryptography secure against any advances in computing technology, provided quantum mechanics remains an accurate description of the laws of nature.

*Authenticated key establishment* (AKE) is the cryptographic task which QKD achieves. The classical cryptographic literature has extensively studied AKE since the founding of public key cryptography in 1976. After a period of ad hoc security analysis of key establishment protocols based on resistance to various individual attacks, protocols are now generally analyzed within the context of a security model, which aims to capture a wide variety of security properties in the context of an attacker who can control all communication, as well as possibly compromise participants; proofs typically consist of probabilistic reductions to computationally hard problems. One seminal model for security of AKE protocols was proposed by Bellare and Rogaway [BR93]. The BR model led to the CK01 model by Canetti and Krawczyk [CK01], upon which was based the eCK model [LLM07]. An alternative approach to this family of security models is given by Canetti’s *universal composability framework* [Can01]. One of the general observations of this line of work has been that calculating a secret key is relatively easy, but properly modelling authentication — ensuring that the key is shared with precisely the intended party and no other — requires greater care.

There are many types of QKD protocols, but for our purposes we will divide them into 3 classes: prepare-send-measure protocols, measure-only protocols, and prepare-send-only protocols. The first QKD protocol, now called BB84, was proposed by Bennett and Brassard [BB84]; it is an example of a prepare-send-measure protocol in which Alice randomly prepares one of several quantum states, sends it to Bob, and Bob randomly measures in one of several settings. Ekert [Eke91] proposed an entanglement-based protocol, which is an example of a measure-only protocol: Alice and Bob only randomly measure in one of several settings; the state itself can be prepared by Eve entirely untrusted. Biham et al. [BHM96] proposed a prepare-send-only protocol, in which Alice and Bob each randomly prepare one of several quantum states and send them to Eve, who measures and sends back a classical result. Different versions can be appealing due to ease of implementation, resistance to side-channel attacks on preparing or measuring, or device independence.

Research arguing for the security of QKD has largely proceeded independent of the aforementioned classical AKE security models. Various proofs of QKD have been given in a stand-alone 2-party setting; some of the most important ones include [May96, LC99, BBB<sup>+</sup>00, SP00, Ina02, GLLP04, Ren05], but many others exist for different variants of QKD; some work on QKD has been done in the universal composability framework [BOHL<sup>+</sup>05]. These proofs typically proceed under the assumption that classical communication happens over an authentic public channel; details on authenticating the classical communication are typically left out of the analysis. It is widely recognized that the authentication can be secure against an unbounded adversary if all classical communication is protected by information-theoretically secure message authentication codes, such as the Wegman-Carter 2-universal hash function [CW79, WC81]. Alternatively, it is generally considered folklore [PPS07, ABB<sup>+</sup>07, SML10, IM11] that if QKD was performed using a computationally secure authentication scheme (such as public key digital signatures), then messages encrypted under the keys output by QKD would be secure provided that the adversary could not break the authentication scheme *before or during* the QKD protocol.

**Contributions.** Our goal is to describe the security of quantum key distribution in a security model similar to existing classical authenticated key exchange protocols and compare the relative security properties of various QKD and classical AKE protocols. Our model is explicitly a multi-party model, includes authentication, and allows for either computationally secure or information theoretically secure authentication. We aim to capture two properties: (1) QKD is *immediately secure* against an active adversary who is restricted such that he is unable to break the authentication scheme, and (2) QKD is *long-term secure*, meaning that, if it is secure against an active adversary who is restricted during the run of the protocol to be unable to break

---

<sup>1</sup>We use the adjective “classical” to mean “non-quantum”, so “classical cryptography” means “non-quantum cryptography”, not “historical cryptography”.

the authentication scheme, then it remains secure even when the (classical and quantum) data obtained by the active bounded adversary are subsequently given to an unbounded quantum adversary.

*Security model for classical-quantum AKE protocols.* In particular, we first introduce in Section 2 a multi-party model for analyzing the security of QKD protocols. In our model, which adopts the formalism of Goldberg et al.’s framework for authenticated key exchange [GSU12], parties consist of a pair of classical and quantum Turing machines, each of which is capable of sending and receiving messages. The adversary controls all communications between parties, but is restricted in its ability to affect communication between a single party’s classical and quantum devices. The adversary also has the ability to compromise various values used by parties during or after the run of the protocol. As is typical, the adversary’s goal is to distinguish the session key of a completed session from a random string of the same length.

Having defined the adversarial model, we then introduce our two security definitions, *immediate security* against an active, potentially bounded adversary, and *long-term security*, meaning security against an adversary who during the run of the protocol is potentially bounded, but after the protocol completes is unbounded (except by the laws of quantum mechanics). Our model is generic enough to allow the bound on the adversary to be computational — assuming that a particular computational problem is hard — or run-time or memory-bounded [CM97]. We adapt the long-term security notion of Müller-Quade and Unruh [MQU10] from the classical universal composability framework to our classical-quantum model.

*Security of BB84.* We then proceed in Section 3 to show that the BB84 protocol, when used with a computationally secure classical authentication scheme such as a digital signature, is secure in this model. For the quantum aspects of the proof, we rely on existing proof techniques, but when combined with the signature scheme in our model, this work provides a proof of the folklore theorem that QKD, when used with computationally secure authentication in a multi-party setting, is information theoretically secure, provided the adversary did not break the authentication during the run of the protocol. Note, importantly, that this is the first proof of QKD in a multi-party setting; while our QKD protocol is still a 2-party protocol, it operates in an environment where many parties may be interacting simultaneously, whereas previous proofs of security of QKD — including the universal composability proof of Ben-Or et al. [BOHL<sup>+</sup>05] — deal with only 2 honest parties (plus the adversary).

*Comparison of quantum and classical AKE protocols.* Finally, we use our generic security model to compare in Section 4 the security properties of classical key exchange protocols and examples from each of the three classes of QKD protocols (prepare-send-measure, measure-only, prepare-send-only). This comparison is facilitated by our phrasing of QKD in a security model more closely related to traditional AKE security models, which we can then use to compare the relative powers afforded to the adversary under those models. In particular, our model allows us to compare how different protocols react when the randomness used in the protocol is revealed — or if it is later discovered that bad randomness was used. For example, some classical AKE protocol such as UP [Ust09] is secure even if the randomness used for either a party’s long-term secret key or ephemeral secret key is revealed *before* the run of the protocol, but the same is not true for the randomness used to pick basis choices in BB84. And the EPR protocol of Ekert is secure even if all of the randomness used by the parties is leaked after the protocol completes, unlike BB84 where data bit choices must remain secret.

## 2 QKD model

Our model begins as an enhancement to the eCK model [LLM07] in which each party has access to a quantum device. The quantum device may be viewed as limited based on for example current hardware limitations. As usual we consider interactive protocols within a multi-party multi-session setting, where communication is controlled by the adversary. Subject to quantum physics restriction the adversary controls the quantum communication channel between parties. Having described the parties and the communication model, we describe how, if at all, the adversary may gain access to secrets used by the parties. We then define secrecy against bounded adversaries and long-term security against unbounded adversaries: the long-term security definition is achieved by having the active bounded short-term adversary output a classical and quantum transcript upon which the unbounded quantum adversary may operate.

We next formally describe the model. We use  $k$  to denote a security parameter. In the description we utilize only qubits, but these can if necessary be generalized to arbitrary-dimension quantum systems.

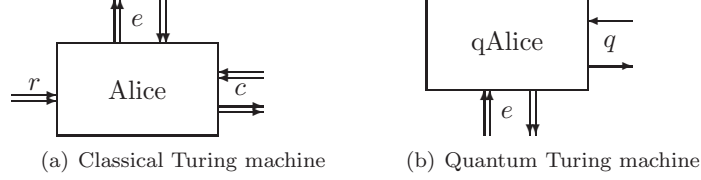


Figure 1: A party's classical and quantum Turing machines

## 2.1 Parties and protocols

**A party** (see also [ABOE10, Definition 1.1 second bullet]) is an interactive classical Turing machine with access to a quantum Turing machine. Typically we refer to this pair of devices jointly as the party.

The classical machine can activate the quantum device via a special activation request or receive (via designated activation routines) measurement outcomes from the quantum device. The communication is delivered over a two way classical communication tape (the  $e$ -channel in Figure 1(a)). The classical Turing machine has also access to a sequence of random bits – the  $r$ -tape in Figure 1(a) – and a separate  $c$ -tape over which the party can receive and send other activation requests and messages as specified by designated routines. Similarly, the quantum Turing device can be activated by the classical Turing machine and can receive and send qubits over a designated quantum channel  $q$  as in Figure 1(b).

Each party can have associated authenticated public strings (which can be public keys or identifiers). Such public strings are assumed to be distributed over an authenticated channels to other parties. Furthermore, pairs of parties may possess shared secrets that were a priori distributed over a confidential and authenticated channel.

**A protocol** is a collection of interactive classical and quantum subroutines that produce a shared secret key between two (or more parties) or output an indicator of an error. The interactions may use messages received on either the classical or quantum channels. The final output of the protocol is made via the classical Turing machine.

**A session** is an execution of the protocol. Sessions are initiated via a special incoming request and upon initiation each one is identified with a unique<sup>2</sup> *session identifier*  $\Psi$  chosen by the party at which the session is executed (in which case we say the party *owns* the session). A session that has been initiated but is not yet completed is called *active*. Since sessions are interactive procedures a party may own more than one active session at a given point of time.

Each active session has a separate *session state* that stores session-specific classical data.<sup>3</sup>

Upon receiving and sending all protocol messages and performing the required measurements and computations specified by the protocol, the session *completes* by having the classical Turing machine output either an error symbol  $\perp$  or a tuple  $(sk, pid, \mathbf{v}, \mathbf{u})$ . The tuple consists of:

- $sk$ : a session key;
- $pid$ : a party identifier;
- $\mathbf{v}$ : a vector  $(\mathbf{v}_0, \mathbf{v}_1, \dots)$  where each  $\mathbf{v}_i$  is a vector of public values or labels; (For example,  $\mathbf{v}_1$  may consist of the public values contributed by party  $P_1$ . Including  $\mathbf{v}$  as part of the session output binds the session with the various values used by the parties to compute the session key.)
- $\mathbf{u}$ : a vector  $(\mathbf{u}_0, \mathbf{u}_1, \dots)$  where each vector  $\mathbf{u}_i$  is a public value or label;  $\mathbf{u}$  is called the *authentication vector* and indicates what information the session owner uses to identify its peer  $pid$ .

<sup>2</sup>With this definition uniqueness is guaranteed only within a party; globally uniqueness can be guaranteed by requiring the session identifier is the concatenation of the unique party identifier and the party's own session identifier

<sup>3</sup>While quantum protocols in general may make use of quantum memory for storing quantum states during a session, the current QKD protocols we consider in this paper, such as BB84 or EPR, do not, so we omit this from our model.

**Definition 1** (Correctness). *A key exchange protocol  $\pi$  is said to be correct if, when all protocol messages are relayed faithfully, without changes to content or ordering, the peer parties output the same session key  $k$  and the same vector  $\mathbf{v}$ .*

**Memory.** A party may hold in its memory several *value pairs* of the form  $(x, X)$ , where  $x$  is a private value and  $X$  is a public value or label. The pair may be a *public key pair*, such as private key  $x$  and public key  $X$ , or a *labelled private value*, such as a private value  $x$  and a unique public label  $X = \ell(x)$ . The value pairs may be generated by some algorithm specified by the protocol.

There are two classifications of value pairs: *ephemeral* value pairs, which are associated with a particular session  $\Psi$ , and *static* value pairs, which can be used across multiple sessions. The party may also have value pairs that have been generated but not yet used. If necessary, different types of key pairs may be permitted, for example, if a protocol uses one type of key pair for digital signatures and another type of key pair for public-key encryption. The protocol specifies an algorithm for generating new pairs.

**Classical Turing machine communication.** As described above each classical Turing machine has two incoming-outgoing classical communication channels, denoted by  $e$  and  $c$  in Figure 1(a), over which the classical Turing machine receives activations and submits responses. The responses themselves can be activation requests. Furthermore the classical Turing machine has an input of classical (pseudo-)random bits which can be read at will by the Turing machine, denoted by  $r$  in Figure 1(a).

The following activations of the classical Turing machine are allowed:

- **SendC( $params, pid$ ):** This activation is received via channel  $c$  and directs the party to begin a new key exchange session. A new session is initiated and assigned a unique session identifier  $\Psi$  based on protocol-specific public parameters  $params$  and an identifier  $pid$  of the party with whom to establish the session. The response to this query includes the session identifier  $\Psi$  and any protocol-specific outgoing classical message  $msg'$  that are sent via the outgoing channel  $c$ . If required by the protocol specification the Turing machine can send an activation request  $C2Q(m)$  over the  $e$  outgoing channel; the activation of the quantum Turing machine may cause that quantum Turing machine to write an output to its  $q$  channel as well, or to prepare its measurement device to receive quantum messages.
- **SendC( $\Psi, msg$ ):** This query models the delivery of classical messages over  $c$ -channel. The party's classical Turing machine is activated with session  $\Psi$  and classical message  $msg$ . It returns any outgoing classical message  $msg'$  over the  $c$ -channel. If required by the protocol specification the Turing machine can send an activation request  $C2Q(m)$  over the  $e$  outgoing channel; the activation of the quantum Turing machine may cause that quantum Turing machine to write an output to its  $q$  channel as well, or to prepare its measurement device to receive quantum messages.
- **Q2C( $m$ ):** Upon activation with this query the classical Turing machine activates its most recent session with input  $m$ . This query may cause the classical Turing machine to output to its  $c$  channel, or send another activation over the  $e$  channel.

A protocol specification may request that the classical Turing machine act probabilistically. In this case the classical machine obtains random bits from the  $r$ -channel.

**Quantum Turing machine communication.** Each party's quantum Turing machine has an incoming-outgoing classical communication channel, denoted by  $q$  in Figure 1(b), over which the machine receives and submits quantum information. The responses themselves can be activation requests. Furthermore the quantum Turing machine has a two-way classical control channel (denoted by  $e$  in Figure 1(b)) with which it communicates with the classical Turing machine.

The following activations of the quantum Turing machine are allowed:

- **SendQ( $\rho$ ):** This query activates the quantum Turing machine with quantum message  $\rho$ ; it returns any outgoing quantum message  $\rho'$  over the  $q$ -channel. If required by the protocol specification the Turing machine can send an activation request  $C2Q(m)$  over the  $e$  outgoing channel, for example, to report any measurement results obtained from measuring  $\rho$ . The activation of the classical Turing machine may cause that classical Turing machine to write an output to its  $c$  channel as well.



- **C2Q( $m$ )**: This query activates the quantum Turing machine with classical control message  $m$ , for example to prepare the quantum circuit for execution due to an anticipated **SendQ** activation. The activation may cause a quantum state to be output over the outgoing quantum channel  $q$  as well as a classical message to be returned over the classical control channel  $e$ .

## 2.2 Adversarial model

**The adversary** is, similar to a party, a pair of interactive classical and quantum Turing machines. The adversary's classical Turing machine runs in time at most  $t_c(k)$  and has access to a quantum Turing machine with runtime bounded by  $t_q(k)$  and memory bounded by  $m_q(k)$  qubits; bounds may be unlimited. The adversary takes as its input all public information and may interact with the (honest) parties. Furthermore the adversary can establish corrupted (dishonest) parties that are fully in control of the adversary. Honest parties are unable to distinguish between honest and dishonest parties.

Communication over the parties' classical  $c$ -channels is controlled by the adversary. On the classical channels, the adversary can read, copy, reorder, insert, delay, modify, drop or forward messages at will. The sending and receiving parties have no intrinsic mechanism to detect which actions, if any, the adversary performed on the classical messages.

Communication over the parties' quantum  $q$  channels is also controlled by the adversary. The adversary's operations on the quantum channels are bound by the laws of quantum mechanics: the delivery of quantum messages can be delayed, modified in order, forwarded, or dropped; the adversary can create new quantum states and perform joint quantum operations on quantum messages received from the parties as well as on the adversary's state. However, due to the laws of quantum mechanics, the adversary cannot necessarily obtain full information about quantum messages from the parties; for example, measurements by the adversary may irrevocably disturb the state of messages transmitted by the parties, and the adversary may be unable to precisely copy a message due to the no-cloning theorem. We assume that the communication channel between the adversary's quantum machine and party's quantum machines are perfect; the adversary's quantum device can simulate any environmental or noise affect on the qubits sent by a party.

**Queries.** The adversary can direct a party to perform certain actions by sending any of the aforementioned activation queries over party's the  $c$  and  $q$  channels. The adversary has neither immediate control and cannot observe the content exchanged between the classical and quantum subcomponents of a party over the  $e$  channel, nor has information about the bits obtained from the  $r$ -channel. Furthermore, to allow for information leakage the adversary may issue the following queries to parties:

- **RevealNext**  $\rightarrow X$ : This query allows the adversary to activate the classical Turing machine to read input from the  $r$ -channel and learn future public values. The activated party generates a new value pair  $(x, X)$ , records it as unused, and returns the public value  $X$ . (This query may be specialized in the event that there are multiple value pair types specified by the protocol.)
- **Partner** $(X) \rightarrow x$ : This query allows the adversary to compromise secret values used in the protocol computation. If the party has a value pair  $(x, X)$  in its memory, it returns the private value  $x$ . **Partner** $(\Psi)$  returns the secret key  $sk$  for session  $\Psi$ , if it exists; this is often referred to as a **RevealSessionKey** query.

Where necessary to avoid ambiguity, we use a superscript to indicate the party to whom the query is directed, for example **SendC** $^{P_i}(\Psi, msg)$ .

**Partnering.** If  $(x, X)$  is a value pair, with public key value or public label  $X$ , then the adversary is said to be a *partner* to  $X$  if the adversary issued the query **Partner** $(X)$  to a party holding that value pair in its memory. Whenever a party generates a key pair  $(x, X)$ , for example in response to a session activation or a **RevealNext** query, the adversary is *not* a partner to  $X$  until the query **Partner** $(X)$  is issued. The adversary can become a partner to any value  $X$ .

## 2.3 Security definition

For the purpose of defining session key security, the adversary has access to the following additional oracle:

- $\text{Test}(i, \Psi) \rightarrow \kappa$ : If party  $P_i$  has not output a session key, return  $\perp$ . Otherwise, choose  $b \xleftarrow{\$} \{0, 1\}$ . If  $b = 1$ , then return the session key  $sk$  from the output for session  $\Psi$  at party  $P_i$ . If  $b = 0$ , return a random bit string of length equal to the length of the session key  $sk$  in session  $\Psi$  at party  $P_i$ . Only one call to the  $\text{Test}$  query is allowed.

**Definition 2** (Fresh session). *A session  $\Psi$  owned by an honest party  $P_i$  is fresh if all of the following occur:*

1. *For every vector  $\mathbf{v}_j$ ,  $j \geq 1$ , in  $P_i$ 's output for session  $\Psi$ , there is at least one element  $X$  in  $\mathbf{v}_j$  such that the adversary is not a partner to  $X$ .*
2. *The adversary did not issue  $\text{Partner}(\Psi')$  to any honest party  $P_j$  for which  $\Psi'$  has the same public output vector as  $\Psi$  (including the case where  $\Psi' = \Psi$  and  $P_j = P_i$ ).*
3. *At the time of session completion, for every vector  $\mathbf{u}_j$ ,  $j \geq 1$ , in  $P_i$ 's output for session  $\Psi$ , there was at least one element  $X$  in  $\mathbf{u}_j$  such that the adversary was not a partner to  $X$ .*

We emphasize the difference between the first and the third condition in the last definition: the latter is decided at the time when the session completes, whereas the former is decided at the end of the adversary's execution. In other words, there may be some values that are okay for the adversary to learn after completion (but not before), and other values that the adversary can never learn.

**Definition 3** (Security). *Let  $k$  be a security parameter. An authenticated key exchange protocol is secure if, for all adversaries  $\mathcal{A}$  with classical runtime bounded by  $t_c(k)$ , quantum runtime bounded by  $t_q(k)$ , and quantum memory bounded by  $m_q(k)$ , the advantage of  $\mathcal{A}$  in guessing the bit  $b$  used in the  $\text{Test}$  query of a fresh session is negligible in the security parameter; in other words, the probability that  $\mathcal{A}$  can distinguish the session key of a fresh session from a random string of the same length is negligible.*

## 2.4 Long-term security

One of the main benefits of quantum key distribution is that it can be secure against unbounded adversaries. However, such strong security comes at the cost of being unable to use computationally secure cryptographic primitives such as public key digital signatures for authentication. The definition above can be used to analyze QKD when computationally secure cryptographic primitives are used; for example, we can choose a  $t_c(k)$ ,  $t_q(k)$ , and  $m_q(k)$  such that the cryptographic primitive is believed secure against such an adversary. The particular values may be chosen based on known classical algorithms for factoring or computing discrete logarithms and on the present-day limits of quantum devices.

Regardless of the bound on the active adversary, we can still recover a very strong form of long-term security by considering an unbounded quantum Turing machine acting after the protocol has completed. In other words, during the run of the protocol, we assume a bounded adversary as in Definition 3; this bounded active adversary produces some classical and quantum transcript which it then provides to the unbounded adversary. This models the real-world scenario of an adversary being somewhat limited by its classical and quantum computing equipment now but later having much more powerful equipment or making an algorithmic breakthrough.

**Definition 4** (Long-term security). *An authenticated key exchange protocol is long-term secure if, for all unbounded quantum Turing machines  $\mathcal{M}$  acting on a classical and quantum transcript produced by a (bounded) adversary  $\mathcal{A}$  in Definition 3, the advantage of  $\mathcal{M}$  in guessing the bit  $b$  used in the  $\text{Test}$  query of a fresh session is negligible in the security parameter.*

## 2.5 Discussion

Several aspects of our model allow for a great range of flexibility in terms of adversarial power and allows quantum key distribution be fairly compared with classical key establishment. We will describe a few specializations of our definition and comment on one of the key differences between our model and traditional classical AKE models, the output vectors  $\mathbf{v}$  and  $\mathbf{u}$ .

**Bounds on devices.** First, if  $t_q(k) = m_q(k) = 0$ , and Definition 4 is omitted, the model reduces to a classical definition for secure session key establishment. It refines the idea of authentication as the session output can explicitly identify how peers were identified and authenticated. Thus any classical protocol analyzed in [GSU12] can also be analyzed in the model presented here. The definition here is stronger in the sense it encompasses a wider range of protocols and relates to the definitions in [LLM07, CK01] the same way [GSU12] relates to them.

Secondly, it is feasible to model present limitations of quantum devices. While there are ongoing improvements in controlling quantum systems, at present the number of qubits a device can work with is essentially a small constant compared to classical computers. Thus, using our model with appropriate values of  $t_q(k)$  and  $m_q(k)$ , based on beliefs about current practical limitations, one can devise efficient protocols that are easy to implement but guarantee unconditional future secrecy. An appropriate assumption on  $t_c(k)$  — for example that all adversaries with polynomial running time  $t_c(k)$  cannot solve a particular hard problem — allow the model to be used as existing classical reductionist security models are used.

Of course, the devices available to the adversary can be made unbounded essentially allowing a complete quantum world. Thus the definitions presented here are suitable for analyzing novel quantum key distribution protocols. These alternatives show the wide range of scenarios our definitions incorporate. Due to the unified underlying framework it is easier to compare various protocols and decide which one is the best for the task at hand.

**The output vectors.** One of the key differences between our model and traditional AKE security models is how we phrase restrictions on what secret values the adversary can learn and when. In the eCK model, for example, a fresh session is defined as one in which the adversary has not learned (a) both the session owner’s ephemeral secret key  $x$  and long-term secret key  $a$ , and (b) both the peer’s ephemeral secret key  $y$  and long-term secret key  $b$  (or just the peer’s long-term key if no matching peer session exists). In our model, this could be specified as  $\mathbf{v} = (\mathbf{v}_0 = (a, x), \mathbf{v}_1 = (b, y))$ .

Since in traditional AKE security models the restriction on values learned is specified in the security model, a new security model is required for each differing combination of learnable values. Though models may often appear similar, they sometimes contain subtle but important formal differences and thus become formally incomparable [Cre11]. The traditional approach of specifying the values that can or cannot be learned in the security definition itself contrasts with our approach — building on that of Goldberg et al. [GSU12] — where the vectors  $\mathbf{v}$  and  $\mathbf{u}$  in the session output specify what can or cannot be learned. As a result, two protocols with differing restrictions on values that can be learned could both be proven secure in our model and then compared based on which values can or cannot be revealed.

### 3 BB84

We now turn to BB84 protocol [BB84]. We first specify the protocol in the language of the model of Section 2, discuss some aspects of our formulation, and complete the section with a security analysis.

**Definition 5.** *Let  $k$  be a security parameter. The BB84 protocol is defined by having parties responding to activations as follows:*

1. Upon activation  $\text{SendC}(\text{start}, \text{initiator}, B)$  the classical Turing machine  $A$  does the following:
  - (a) create a new session  $\Psi^A$  with peer identifier  $B$ ;
  - (b) read  $n_1$  (random) data bits  $\Psi_{dAB}^A$  from its  $r$ -tape;
  - (c) read  $n_1$  (random) basis bits  $\Psi_{bA}^A$  from its  $r$ -tape;
  - (d) send activation  $\text{C2Q}(\Psi_{bA}^A, \Psi_{dAB}^A)$  on its  $e$ -tape, which indicates that the quantum device should encode each data bit from  $\Psi_{dAB}^A$  as  $|0\rangle$  or  $|1\rangle$  if the corresponding basis bit  $\Psi_{bA}^A$  is 0, or as  $|+\rangle$  or  $|-\rangle$  if the corresponding basis bit  $\Psi_{bA}^A$  is 1;
  - (e) send activation  $\text{SendC}(\Psi^A, \text{start}, \text{responder}, A)$  on its  $c$ -tape to  $B$ .
2. Upon activation  $\text{SendC}(\Psi^A, \text{start}, \text{responder}, A)$  the classical Turing machine  $B$  does the following:



- (a) create a new session  $\Psi^B$  with peer identifier  $A$ ;
  - (b) read  $n_1$  (random) basis bits  $\Psi_{bB}^B$  from its  $r$ -tape;
  - (c) send activation  $\text{C2Q}(\Psi_{bB}^B)$  on its  $e$ -tape, which indicates the quantum device should measure the  $i$ th qubit in the  $|0\rangle/|1\rangle$  if the  $i$ th bit of  $\Psi_{bB}^B$  is 0, or in the  $|+\rangle/|-\rangle$  basis if  $i$ th bit of  $\Psi_{bB}^B$  is 1.
3. Upon activation  $\text{Q2C}(m)$ , the classical Turing machine  $B$  does the following:
  - (a) set  $\Psi_{dAB}^B$  equal to  $m$ ;
  - (b) compute  $\sigma \leftarrow \text{Sign}_{pk_B}(\Psi^A, \Psi^B, \Psi_{bB}^B, B)$ ;
  - (c) send activation  $\text{SendC}(\Psi^A, \Psi^B, \Psi_{bB}^B, \sigma)$  on its  $c$ -tape to  $A$ .
4. Upon activation  $\text{SendC}(\Psi^A, \Psi^B, \Psi_{bB}^B, \sigma)$ , the classical Turing machine  $A$  does the following:
  - (a) verify  $\sigma$  with  $pk_B$ ;
  - (b) discard all bit positions from  $\Psi_{dAB}^A$  for which  $\Psi_{bA}^A$  is not equal to  $\Psi_{bB}^B$ ; assume there are  $n_2$  such positions left;
  - (c) read  $n_2$  (random) bits  $\Psi_{indAB}^A$  from its  $r$ -tape; set  $\Psi_{chkAB}^A$  to be the substring of  $\Psi_{dAB}^A$  for which the bits of  $\Psi_{indAB}^A$  are 1, and set  $\Psi_{kAB}^A$  to be the substring of  $\Psi_{dAB}^A$  for which the bits of  $\Psi_{indAB}^A$  are 0; let  $n_3$  denote the length of  $\Psi_{kAB}^A$ ;
  - (d) compute  $\sigma \leftarrow \text{Sign}_{pk_A}(\Psi^A, \Psi^B, \Psi_{bA}^A, \Psi_{indAB}^A, \Psi_{chkAB}^A, A)$ ;
  - (e) send activation  $\text{SendC}(\Psi^A, \Psi^B, \Psi_{bA}^A, \Psi_{indAB}^A, \Psi_{chkAB}^A, \sigma)$  on its  $c$ -tape to  $B$ .
5. Upon activation  $\text{SendC}(\Psi^A, \Psi^B, \Psi_{indAB}^A, \Psi_{chkAB}^A, \sigma)$ , the classical Turing machine  $B$  does the following:
  - (a) verify  $\sigma$  with  $pk_A$ ;
  - (b) discard all bit positions from  $\Psi_{dAB}^B$  for which  $\Psi_{bA}^A$  is not equal to  $\Psi_{bB}^B$ ;
  - (c) set  $\Psi_{chkAB}^B$  to be the substring of  $\Psi_{dAB}^B$  for which the bits of  $\Psi_{indAB}^A$  are 1, and set  $\Psi_{kAB}^B$  to be the substring of  $\Psi_{dAB}^B$  for which the bits of  $\Psi_{indAB}^A$  are 0;
  - (d) let  $\epsilon$  be the proportion of bits of  $\Psi_{chkAB}^B$  that do not match  $\Psi_{chkAB}^A$ ; if  $\epsilon > 0.061$  then abort;
  - (e) compute  $\sigma \leftarrow \text{Sign}_{pk_B}(\Psi^A, \Psi^B, \epsilon, B)$ ;
  - (f) send activation  $\text{SendC}(\Psi^A, \Psi^B, \epsilon, \sigma)$  on its  $c$ -tape to  $A$ .
6. Upon activation  $\text{SendC}(\Psi^A, \Psi^B, \epsilon, \sigma)$ , the classical Turing machine  $A$  does the following:
  - (a) verify  $\sigma$  with  $pk_B$ ;
  - (b) read (random) bits  $\Psi_F^A$  from its  $r$ -tape to construct a random a 2-universal hash function  $F : \{0, 1\}^{n_3} \rightarrow \{0, 1\}^{r'}$  (where  $r' = n_3 h(\epsilon) + o(n_3)$ ) for information reconciliation (see Appendix A) and compute  $F' = F(\Psi_{kAB}^A)$ ;
  - (c) read (random) bits  $\Psi_{P,G}^A$  from its  $r$ -tape to generate a random permutation  $P$  on  $n_3$  elements and a 2-universal hash function  $G : \{0, 1\}^{n_3} \rightarrow \{0, 1\}^{s'}$  (where  $s' = n_3(1 - 3h(\epsilon)) + o(n_3)$ ) for privacy amplification (see Appendix A), respectively; compute  $\Psi_{skAB}^A \leftarrow G(P(\Psi_{kAB}^A))$ ;
  - (d) compute  $\sigma \leftarrow \text{Sign}_{pk_A}(\Psi^A, \Psi^B, F, F', P, G, A)$ ;
  - (e) send activation  $\text{SendC}(\Psi^A, \Psi^B, F, F', P, G, \sigma)$  on its  $c$ -tape to  $B$ ;
  - (f) output  $(sk = \Psi_{skAB}^A, pid = B, \mathbf{v} = (\mathbf{v}_0 = (\ell(\Psi_{dAB}^A)), \mathbf{v}_1 = (\ell(\Psi_{bAB}^A)), \mathbf{v}_2 = (\ell(\Psi_{dAB}^B)), \mathbf{v}_3 = (\ell(\Psi_{bAB}^B)), \mathbf{v}_4 = (\ell(\Psi_F^A)), \mathbf{v}_5 = (\ell(\Psi_{P,G}^A))), \mathbf{u} = (\mathbf{u}_1 = (pk_B)))$  (recall  $\ell(\cdot)$  denotes the label describing the corresponding secret value).
7. Upon activation  $\text{SendC}(\Psi^A, \Psi^B, F, F', P, G, \sigma)$ , the classical Turing machine  $B$  does the following:
  - (a) verify  $\sigma$  with  $pk_A$ ;
  - (b) use  $F$  and  $F'$  to correct  $\Psi_{kAB}^B$  to  $\Psi_{kAB'}^B$ ;

- (c) compute  $\Psi_{skAB}^B \leftarrow G(P(\Psi_{kAB'}^B))$ ;  
(d) output  $(sk = \Psi_{skAB}^B, pid = A, \mathbf{v} = (\mathbf{v}_0 = (\ell(\Psi_{dAB}^A)), \mathbf{v}_1 = (\ell(\Psi_{bAB}^A)), \mathbf{v}_2 = (\ell(\Psi_{dAB}^B)), \mathbf{v}_3 = (\ell(\Psi_{bAB}^B)), \mathbf{v}_4 = (\ell(\Psi_F^A)), \mathbf{v}_5 = (\ell(\Psi_{P,G}^A)), \mathbf{u} = (\mathbf{u}_1 = (pk_A)))$ .

**Remark 1.** In the output vector  $\mathbf{v}$ , the values  $\ell(\Psi_{bAB}^A)$ ,  $\ell(\Psi_{bAB}^B)$ ,  $\ell(\Psi_F^A)$ , and  $\ell(\Psi_{P,G}^A)$  appear as single component vectors. But in step 6(e) the values are broadcast in the clear. This may seem a bit contradictory since, if the adversary becomes a partner to either of those values (and therefore learns their content), the session is not fresh, but because of the broadcast the adversary does in fact learn the values corresponding to the aforementioned labels. The important distinction is when the adversary obtains these values, either before or after the protocol commences and measurements are performed. For the adversary to learn these values before parties' measurements, it must partner to these values, violating session freshness. Learning the values after the session completes is not an issue and the values are given to the adversary "for free", without the need for partnering.

**Remark 2.** The output vector  $\mathbf{u}$  represents the values which the session owner uses to authenticate its peer. Similar to  $\ell(\Psi_{bAB}^A)$  the authentication information has to be exclusively available to the alleged peer, but only at the time of protocol execution: these values may subsequently be revealed. Therefore, as described in Definition 2, partnering to  $\mathbf{u}$  is decided upon session completion.

Observe that for the BB84 protocol above, Alice does not include her own authentication secret  $pk_A$ . This implies that the protocol is resilient to key compromise impersonation attacks: even with Alice's authentication keys no party is able to pretend to be someone other than Alice to Alice.

### 3.1 Security of BB84

We now show that the BB84 protocol stated above is a secure and long-term-secure authenticated key exchange protocol under the assumption that the bounded active adversary cannot break the signature scheme.

**Theorem 1** (Security of BB84). *Let  $k$  be a security parameter. Suppose that the probability  $\epsilon_{\text{sig}}$  that any probabilistic polynomial time classical Turing machine with oracle access to a  $(t_q(k), m_q(k))$ -bounded quantum Turing machine can break the signature scheme is negligible in  $k$ . Then the BB84 protocol is a secure authenticated key exchange protocol according to Definition 3.*

*Proof sketch.* Our proof combines an existing proof of security by Christandl et al. [CRE04] for the BB84 protocol with the sequence-of-games technique of Shoup [Sho06]. First we show — using techniques from classical reductionist security — that no bounded adversary can (except with negligible probability) successfully tamper with the classical authenticated communication. Then we show — using techniques from QKD security proofs — that the adversary cannot distinguish the key from random. Details appear in Appendix B.

**Theorem 2** (Long-term security of BB84). *Let  $k$  be a security parameter. Suppose the signature scheme is secure against all bounded adversaries as specified in Theorem 1. Then the BB84 protocol is a long-term secure authenticated key exchange protocol according to Definition 4.*

*Proof.* The argument has in fact appeared in the argument of Theorem 1. Observe that in its proof the bounds on  $t_c(k)$ ,  $t_q(k)$ , and  $m_q(k)$  and on the adversary general is required only for guaranteeing the authenticity and origin of messages in the hop from game 0 to game 1. Once assured that the classical authentic communication has not been tampered with the remainder of the argument is a typical argument for a quantum key distribution scheme, which does not require any bounds on the adversarial power. Since the unbounded adversary runs after the protocol completes, meaning it cannot inject reorder or modify messages in the transcript, the past classical communication remains authentic and the result follows.  $\square$

## 4 Comparing classical and quantum key exchange protocols

In Section 2.5, we discussed how our model can be used to analyze both purely classical protocols and quantum protocols. Given its similarity to existing classical AKE security models and its flexibility in

Table 1: Comparison of security properties of various classical and quantum AKE protocols.

Protocol	Signed Diffie–Hellman [CK01]	UP [Ust09]	BB84 [BB84]	EPR [Eke91]	BHM96 [BHM96, Ina02]
Protocol type	classical	classical	quantum prepare-send-measure	quantum measure-only	quantum prepare-send-only
Security model in which can be proven secure	CK01 [CK01], this paper	eCK [LLM07], this paper	this paper	this paper	this paper
Randomness revealable <b>before</b> protocol run?	× static key × ephemeral key	at most 1 of static key, ephemeral key	× static key × basis choice × data bits × info. recon. × priv. amp.	× static key × basis choice  × info. recon. × priv. amp.	× static key × basis choice × data bits × info. recon. × priv. amp.
Randomness revealable <b>after</b> protocol run?	✓ static key × ephemeral key	at most 1 of static key, ephemeral key	✓ static key ✓ basis choice × data bits ✓ info. recon. ✓ priv. amp.	✓ static key ✓ basis choice  ✓ info. recon. ✓ priv. amp.	✓ static key ✓ basis choice × data bits ✓ info. recon. ✓ priv. amp.
Short-term security	computational assumption	computational assumption	computational or information-theoretic	computational or information-theoretic	computational or information-theoretic
Long-term security	×	×	assuming short-term-secure authentication	assuming short-term-secure authentication	assuming short-term-secure authentication

analyzing the security of a variety of protocols, it is natural to use the model to try to identify qualitative differences between different classes of protocols.

One of the key differences between existing AKE security models such as CK01 and eCK is what randomness the adversary is allowed reveal, and when, and still have the protocol be secure. Our framework is more generic: it is not the *model* that specifies which randomness can be revealed but the *protocol itself* in its output vectors  $\mathbf{v}$  and  $\mathbf{u}$ . As a result, we can “compare” protocols by viewing them all within our model and then comparing which values are included in the output vector.<sup>4</sup>

Table 1 summarizes the observations of this section. We compare are two qualitatively different classical AKE protocols and three qualitatively different QKD protocols: (1) the signed Diffie–Hellman protocol [CK01] (which can be proven secure in the CK01 model), (2) the UP protocol [Ust09], a variant of the MQV protocol [LMQ<sup>+</sup>03] which can be proven secure in the eCK model, (3) the BB84 [BB84] prepare-send-measure QKD protocol, (4) the EPR [Eke91] (entanglement-based) measure-only QKD protocol, and (5) the BHM96 [BHM96, Ina02] prepare-send-only QKD protocol. We note that our model is flexible enough to allow all these protocols to be proven secure in it, of course with different cryptographic assumptions, bounds on the adversary, and different output vectors. It is these differences we compare in Table 1

*Revealing randomness before the run of the protocol.* Some classical AKE protocols, especially eCK-secure protocols such as UP and similar MQV-style protocols, remain secure even if the adversary learns either the ephemeral secret key or the long-term secret key, but not both, before the run of the protocol. This contrasts with all known QKD protocols, where none of the random values used during the protocol — the long-term secret key, the basis choices (for measure protocols), data bits (for prepare protocols), information reconciliation function, or privacy amplification function — can be revealed to the adversary in advance. (This is why all of these values are included individually in the output vector  $\mathbf{v}$  in the BB84 specification in Section 3.)

*Revealing randomness after the run of the protocol.* For classical AKE protocols to remain secure, at least some secret values must not be revealed after the run of the protocol. For protocols with so-called perfect forward secrecy, such as signed Diffie–Hellman, the parties’ long-term secret keys can be corrupted after the run of the protocol, but not the ephemeral secret keys. For eCK-secure protocols such as UP and similar MQV-style protocols, either the long-term secret key or the ephemeral secret key, but not both, can be revealed after the protocol run (or, as per the previous paragraph, before/during). For measure-only entanglement-based QKD protocols such as EPR, all random choices made by the parties can be revealed after the run of the protocol: this is because the key bits are not chosen by the parties, nor in fact by the adversary, but are the result of measurements and (after successful privacy amplification) are uncorrelated

<sup>4</sup>We note that it has been shown [Cre11] that the CK01 and eCK models are *formally incomparable*, meaning neither can be shown to imply the other. Nonetheless, properties of *specific protocols* secure in those models may be compared by resorting to a third model such as the one in this paper.

with any of the input bits of any of the parties, including the adversary. This is not the case for prepare-and-send protocols such as BB84 or BHM96, as the sender does randomly choose data bits which must remain secret.

*Short-term and long-term security.* Classical AKE protocols can be proven secure only under computational assumptions, and as such only offer short-term security in the sense of Definition 3. Even against an unbounded passive adversary they do not retain any of their secrecy properties. Thus classical AKE protocols are only secure against bounded short-term adversaries; however, they can be compared on the relative strength of the bound on the adversary. This contrasts with QKD protocols. QKD can be shown to be secure against either *unbounded* short-term adversaries, by using information-theoretic authentication, or secure against bounded short-term adversaries when using a computationally secure authentication scheme as we have shown for BB84 in Section 3.1. A key contribution of the model in Section 2 is a formalism which captures the notion that QKD can remain secure against an unbounded adversary after the protocol completes, provided the adversary at the time of the run of the protocol could not break the authentication scheme.

We note that applications wishing to achieve both the long-term security properties of QKD and the resistance to randomness revelation that eCK-secure classical AKE protocols have could do so by running both protocols in parallel for each session, and then combining the keys output by the two protocols together; if combined correctly, the resulting key would provide strong short-term security and strong long-term security. This approach is indeed being used by QKD implementers, such as commercial QKD vendor ID Quantique.<sup>5</sup>

## 5 Conclusions

We have presented a model for key establishment which incorporates both classical key agreement and quantum key distribution. Our model can accommodate a wide range of practical and theoretical scenarios and can serve as a common framework in which to compare relative security properties of different protocols. A key aspect of our model is that restrictions on values that the adversary can compromise are not specified by the model but by the output of the protocol. Using our model, we were able to provide a formal argument for the short-term and long-term security of BB84 in the multi-user setting while using computationally secure authentication.

The ability to compare various classical and quantum protocols in our model has allowed us to identify an important distinction between existing classical key establishment and quantum key distribution protocols. At a high level, classical protocols can provide more assurances against online adversaries who can leak or infiltrate in certain ways, but in the long run may be insecure against potential future advances. Current quantum protocols provide assurances against somewhat weaker online adversaries but retain secrecy indefinitely, even against future advances in computing technology.

Since in our model the relative strength of a fresh session is specified by the conditions given in the output vector, an interesting open problem would be to use our model develop a quantum key distribution protocol which does retain its security attributes in the short- and long-terms even if some random values were known before the run of the protocol. Also of interest is how to best combined keys from both quantum and classical key exchange protocols run in parallel.

## Acknowledgements

The authors gratefully acknowledge helpful discussions with Norbert Lütkenhaus, Alfred Menezes, and Kenny Paterson.

MM is supported by NSERC (Discovery, SPG FREQUENCY, CREATE), QuantumWorks, MITACS, CIFAR, ORF. IQC and Perimeter Institute are supported in part by the Government of Canada and the Province of Ontario.

---

<sup>5</sup><http://www.idquantique.com/images/stories/PDF/cerberis-encryptor/cerberis-specs.pdf>

## References

- [ABB<sup>+</sup>07] Romain Alléaume, Jan Bouda, Cyril Branciard, Thierry Debuisschert, Mehrdad Dianati, Nicolas Gisin, Mark Godfrey, Philippe Grangier, Thomas Länger, Anthony Leverrier, Norbert Lütkenhaus, Philippe Painchault, Momtchil Peev, Andreas Poppe, Thomas Pornin, John Rarity, Renato Renner, Grégoire Ribordy, Michel Riguidel, Louis Salvail, Andrew Shields, Harald Weinfurter, and Anton Zeilinger. SECOQC white paper on quantum key distribution and cryptography, January 2007. <http://www.arxiv.org/abs/quant-ph/0701168>.
- [ABOE10] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In Andrew Chi-Chih Yao, editor, *Proc. Innovations in Computer Science (ICS) 2010*, pp. 453–469, October 2010. EPRINT [arXiv:0810.5375](https://arxiv.org/abs/0810.5375), URL <http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/35.html>.
- [BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: public key distribution and coin tossing. In *Proc. IEEE International Conf. on Computers, Systems and Signal Processing*, pp. 175–179. IEEE, December 1984. URL <http://www.research.ibm.com/people/b/bennetc/bennettc198469790513.pdf>.
- [BBB<sup>+</sup>00] Eli Biham, Michel Boyer, P. Oscar Boykin, Tal Mor, and Vwani Roychowdhury. A proof of the security of quantum key distribution (extended abstract). In *Proc. 32nd Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 715–724. ACM Press, 2000. DOI:10.1145/335305.335406. EPRINT [arXiv:quant-ph/9912053](https://arxiv.org/abs/quant-ph/9912053).
- [BHM96] Eli Biham, Bruno Huttner, and Tal Mor. Quantum cryptographic network based on quantum memories. *Physical Review A*, **54**(4):2651–2658, 1996. DOI:10.1103/PhysRevA.54.2651.
- [BOHL<sup>+</sup>05] Michael Ben-Or, M. Horodecki, Debbie W. Leung, Dominic Mayers, and Jonathan Oppenheim. The universal composable security of quantum key distribution. In Joe Kilian, editor, *Theory of Cryptography Conference (TCC) 2005, LNCS*, volume 3378, pp. 386–406. Springer, 2005. DOI:10.1007/b106171. EPRINT [arXiv:quant-ph/0409078](https://arxiv.org/abs/quant-ph/0409078).
- [BR93] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – Proc. CRYPTO ’93, LNCS*, volume 773, pp. 232–249. Springer, 1993. DOI:10.1007/3-540-48329-2\_21. Full version available at <http://www-cse.ucsd.edu/~mihir/papers/key-distribution.html>.
- [Can01] Ran Canetti. Universally composable security: a new paradigm for cryptographic protocols (extended abstract). In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2001*, pp. 136–145. IEEE Press, 2001. DOI:10.1109/SFCS.2001.959888.
- [CK01] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology – Proc. EUROCRYPT 2001, LNCS*, volume 2045, pp. 453–474. Springer, 2001. DOI:10.1007/3-540-44987-6\_28. Full version available at <http://eprint.iacr.org/2001/040>.
- [CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – Proc. CRYPTO ’97, LNCS*, volume 1297, pp. 292–306. Springer, 1997. DOI:10.1007/BFb0052243.
- [CRE04] Matthias Christandl, Renato Renner, and Artur Ekert. A generic security proof for quantum key distribution, February 2004. <http://arxiv.org/abs/quant-ph/0402131v2>.
- [Cre11] Cas Cremers. Examining indistinguishability-based security models for key exchange protocols: the case of CK, CK-HMQV, and eCK. In *Proc. 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS) 2011*, pp. 80–91. ACM, 2011. DOI:10.1145/1966913.1966925.



- [CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, **18**(2):143–154, 1979. DOI:[10.1016/0022-0000\(79\)90044-8](https://doi.org/10.1016/0022-0000(79)90044-8).
- [DHKP97] Martin Dietzfelbinger, Torben Hagerup, Jyrki Katajainen, and Martti Penttonen. A reliable randomized algorithm for the closest-pair problem. *Journal of Algorithms*, **25**(1):19–51, 1997. DOI:[10.1006/jagm.1997.0873](https://doi.org/10.1006/jagm.1997.0873).
- [Eke91] Artur K. Ekert. Quantum cryptography based on Bell’s theorem. *Physical Review Letters*, **67**:661–663, August 1991. DOI:[10.1103/PhysRevLett.67.661](https://doi.org/10.1103/PhysRevLett.67.661).
- [GLLP04] Daniel Gottesman, Hoi-Kwong Lo, Norbert Lütkenhaus, and John Preskill. Security of quantum key distribution with imperfect devices. *Quantum Information and Computation*, **4**(5):325–360, September 2004. EPRINT [arXiv:quant-ph/0212066](https://arxiv.org/abs/quant-ph/0212066), URL <http://www.rinton.net/xqic4/qic-4-5/325-360.pdf>.
- [GSU12] Ian Goldberg, Douglas Stebila, and Berkant Ustaoglu. Anonymity and one-way authentication in key exchange protocols. *Designs, Codes and Cryptography*, 2012. DOI:[10.1007/s10623-011-9604-z](https://doi.org/10.1007/s10623-011-9604-z). URL <http://www.cacr.math.uwaterloo.ca/techreports/2011/cacr2011-11.pdf>. Online first; print version to appear.
- [IM11] Lawrence M. Ioannou and Michele Mosca. A new spin on quantum cryptography: Avoiding trapdoors and embracing public keys. In Bo-Yin Yang, editor, *Proc. 4th International Workshop on Post-Quantum Cryptography (PQCrypto) 2011*, LNCS, volume 7071, pp. 255–274. Springer, 2011. DOI:[10.1007/978-3-642-25405-5\\_17](https://doi.org/10.1007/978-3-642-25405-5_17). EPRINT [arXiv:1109.3235v1](https://arxiv.org/abs/1109.3235v1).
- [Ina02] Hitoshi Inamori. Security of practical time-reversed EPR quantum key distribution. *Algorithmica*, **34**(4):340–365, 2002. DOI:[10.1007/s00453-002-0983-4](https://doi.org/10.1007/s00453-002-0983-4).
- [LC99] Hoi-Kwong Lo and H. F. Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, **283**(5410):2050–2056, 1999. DOI:[10.1126/science.283.5410.2050](https://doi.org/10.1126/science.283.5410.2050). EPRINT [arXiv:quant-ph/9803006](https://arxiv.org/abs/quant-ph/9803006).
- [LLM07] Brian LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *First International Conference on Provable Security (ProvSec) 2007*, LNCS, volume 4784, pp. 1–16. Springer, 2007. DOI:[10.1007/978-3-540-75670-5\\_1](https://doi.org/10.1007/978-3-540-75670-5_1).
- [LMQ+03] Laurie Law, Alfred Menezes, Minghua Qu, Jerry Solinas, and Scott Alexander Vanstone. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, **28**(2):119–134, 2003. DOI:[10.1023/A:1022595222606](https://doi.org/10.1023/A:1022595222606).
- [May96] Dominic Mayers. Quantum key distribution and string oblivious transfer in noisy channels. In Neal Koblitz, editor, *Advances in Cryptology – Proc. CRYPTO ’96*, LNCS, volume 1109, pp. 343–357. Springer, 1996. DOI:[10.1007/3-540-68697-5\\_26](https://doi.org/10.1007/3-540-68697-5_26). EPRINT [arXiv:quant-ph/9802025](https://arxiv.org/abs/quant-ph/9802025).
- [MQU10] Jörn Müller-Quade and Dominique Unruh. Long-term security and universal composability. *Journal of Cryptology*, **23**(4):594–671, 2010. DOI:[10.1007/s00145-010-9068-8](https://doi.org/10.1007/s00145-010-9068-8).
- [PPS07] Kenneth G. Paterson, Fred Piper, and Rüdiger Schack. Quantum cryptography: A practical information security perspective. In Marek Zukowski, Sergei Kilin, and Janusz Kowalik, editors, *Proc. NATO Advanced Research Workshop on Quantum Communication and Security*, NATO Science for Peace and Security Series, Sub-Series D: Information and Communication Security, volume 11. IOS Press, 2007. See also <http://arxiv.org/abs/quant-ph/0406147>.
- [Ren05] Renato Renner. *Security of Quantum Key Distribution*. PhD thesis, Swiss Federal Institute of Technology Zürich, 2005. EPRINT [arXiv:quant-ph/0512258v2](https://arxiv.org/abs/quant-ph/0512258v2).

- [Sho06] Victor Shoup. Sequences of games: A tool for taming complexity in security proofs. <http://www.shoup.net/papers/games.pdf>, 2006. First version appeared in 2004.
- [SML10] Douglas Stebila, Michele Mosca, and Norbert Lütkenhaus. The case for quantum key distribution. In Alexander Sergienki, Saverio Pascazio, and Paolo Villoresi, editors, *Quantum Communication and Quantum Networking: First International Conference, QuantumComm 2009, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, volume 36, pp. 283–296. Springer, 2010. DOI:10.1007/978-3-642-11731-2\_35. EPRINT [arXiv:0902.2839](http://arxiv.org/abs/0902.2839), <http://eprint.iacr.org/2009/082>.
- [SP00] Peter Shor and John Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, **85**(2):441–444, 2000. DOI:10.1103/PhysRevLett.85.441.
- [Ust09] Berkant Ustaoglu. Comparing sessionstatereveal and ephemeralkeyreveal for diffie-hellman protocols. In Josef Pieprzyk and Fangguo Zhang, editors, *Provable Security: Third International Conference, ProvSec 2009, LNCS*, volume 5848, pp. 183–197. Springer, 2009. DOI:10.1007/978-3-642-04642-1\_16.
- [WC81] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, **22**(3):265–279, 1981. DOI:10.1016/0022-0000(81)90033-7.

## A Information reconciliation and privacy amplification

See [CRE04, §4.4.1–4.4.3] for a formal analysis of information reconciliation and privacy amplification in the context of quantum key distribution.

### A.1 2-universal hash functions

A family of *2-universal hash functions* is a set of hash functions  $\mathcal{H}$  mapping a set  $U$  to bit strings of length  $r'$  if, for all  $x, y \in U$  with  $x \neq y$ ,

$$\Pr_{H \in \mathcal{H}} (H(x) = H(y)) \leq 2^{-r'}.$$

An example of a 2-universal hash function is as follows. Fix  $r'$ . Let  $U = \{0, 1, \dots, 2^w - 1\}$ , with  $w > r'$ . Let  $a$  be a randomly chosen positive odd integer with  $a < 2^w$  and let  $b = i2^{w/2}$  where  $i$  is chosen at random from  $\{0, \dots, 2^{w/2} - 1\}$ . Define

$$H_{a,b}(x) = ((ax + b) \bmod 2^w) \operatorname{div} 2^{w-r'}$$

where  $\operatorname{div}$  denotes integer division. Then  $\mathcal{H} = \{H_{a,b} : a, b \text{ as above}\}$  is a family of 2-universal hash functions [DHKP97].

### A.2 Using 2-universal hash functions for information reconciliation

Let  $\epsilon$  be the proportion of Bob's check bits  $ckb_B$  that disagree with Alice's check bits  $ckb_A$ . Set  $r = \lceil nh(\epsilon) \rceil$ . Choose  $r' = r + o(n)$ . Choose  $F \in_R \mathcal{H}_{r'}$ . Alice sends to Bob the description of the function  $F$  and the value  $F(\Psi_{kAB}^A)$ . Bob corrects  $\Psi_{kAB}^B$  to  $\Psi_{kAB}^A$  by guessing the errors and checking based on the received value  $F(\Psi_{kAB}^A)$ .

Note that information reconciliation can also be achieved using certain types of error correcting codes.

### A.3 Using 2-universal hash functions for privacy amplification

Alice chooses a random permutation  $P$  on  $|\Psi_{kAB}^A|$  elements. She also chooses a random 2-universal hash function  $G$  that maps  $|\Psi_{kAB}^A|$  bits to  $s' = n - \lceil 3nh(\epsilon) \rceil + o(n)$  bits, where  $\epsilon$  is (as before) the proportion of Bob's check bits  $ckb_B$  that disagree with Alice's check bits  $ckb_A$ . Alice sends  $P$  and  $G$  to Bob authentically, for example by signing it. Alice computes her final session key as  $\Psi_{skAB}^A = G(P(\Psi_{kAB}^A))$  and Bob computes his final session key as  $\Psi_{skAB}^B = G(P(\Psi_{kAB}^B))$ .

## B Proof of Theorem 1

*Proof.* Our proof combines an existing proof of security by Christandl et al. [CRE04] for the BB84 protocol with the sequence-of-games technique of Shoup [Sho06].<sup>6</sup> First we show — using techniques from classical reductionist security — that no bounded adversary can (except with negligible probability) successfully tamper with the classical authenticated communication. Then we show — using techniques from QKD security proofs — that the adversary cannot distinguish the key from random.

Let  $\text{Succ}_i$  denote the event that, in game  $i$ , the adversary successfully guesses the bit  $b$  used in the Test query against a fresh session.

**Game 0.** This is the original security experiment. Our goal is to prove an upper bound on  $|\Pr(\text{Succ}_0) - \frac{1}{2}|$ .

**From Game 0 to Game 1.** In this game, we want to ensure that all parties that output session keys receive as input over the classical  $c$ -channel exactly the messages sent by its peer’s session. We make use of the fact that each party chooses its session identifier uniquely (within itself) and that these session identifiers are included in every digital signature.

Let  $\text{abort}_{\text{sig}}$  be the event that there exists an honest party  $P$  owning a fresh session  $\Psi^A$  that output a session key such that

- party  $P$  received  $\Psi^B$  as the session identifier of the peer’s session,
- there there is no honest party  $P'$  with session identifier  $\Psi^B$  and peer session identifier  $\Psi^A$ ,
- but the signature received by party  $P$  in either step 4(a) (if  $P$  is “Alice”), step 5(a) (if  $P$  is “Bob”), 6(a) (if  $P$  is “Alice”), or 7(a) (if  $P$  is “Bob”) verifies correctly under the long-term public key of the party corresponding to the peer identifier  $pid$  of the session.

If  $\text{abort}_{\text{sig}}$  occurs, the challenger aborts.

We have that  $|\Pr(\text{Succ}_0) - \Pr(\text{Succ}_1)| \leq \Pr(\text{abort}_{\text{sig}})$ .

We now need a bound on  $\Pr(\text{abort}_{\text{sig}})$ . We will obtain such a bound by constructing a signature forger as follows. The forger receives as input a public key  $pk^*$  and simulates the challenger for the adversary. The challenger guesses an index  $i^*$  of a party, and generates all public keys / secrets for all parties except party  $i^*$  as before. The challenger then proceeds exactly as in game 0, except that whenever party  $P_{i^*}$  is required to generate a signature on a message  $m$ , the challenger uses the signing oracle of the signature challenger.

Suppose event  $\text{abort}_{\text{sig}}$  occurs at some party  $P_i$  in a session with peer identifier  $P_j$ . This means that party  $P_i$  has received as input a signature on either  $(\Psi^A, \Psi^B, \Psi_{bB}^B, B)$  (in step 4(a)),  $(\Psi^A, \Psi^B, \Psi_{bA}^A, \Psi_{indAB}^A, \Psi_{chkAB}^A, A)$  (in step 5(a)),  $(\Psi^B, \Psi^A, \epsilon, B)$  (in step 6(a)), or  $(\Psi^A, \Psi^B, F, F', P, G, A)$  (in step 7(a)) but no session at party  $P_j$  ever issued the corresponding signature, since there is session with identifier  $\Psi^B$  at  $P_j$ .

If  $i^* = j$ , which happens with non-negligible probability  $1/n_P$ , where  $n_P$  is the number of parties, then the forger can use the signature  $\sigma$  received by  $P_i$  to break the existential-unforgeability-under-chosen-message-attack of the signature scheme with success probability at least  $4\epsilon_{\text{sig}}$ . Thus,

$$|\Pr(\text{Succ}_0) - \Pr(\text{Succ}_1)| \leq 4 \cdot n_P \cdot \epsilon_{\text{sig}} . \quad (1)$$

**Game 1.** Having assured that classical communication is untampered with in honest sessions that output session keys, we now make use of standard proofs for security of quantum key distribution. In particular, we follow the technique of Christandl et al. [CRE04]. We provide a brief sketch of their argument.

Let  $\epsilon \geq 0$  be negligible in  $k$ . The  $\epsilon$ -smooth Rényi entropy of order  $\infty$  of a probability distribution  $P$  is denoted by  $H_\infty^\epsilon(P)$  [CRE04, Definition 3.5]. For intuitive purposes, we will refer to this as simply the “sR-entropy” of  $P$ ; the detailed analysis appears in [CRE04, §4.4–5.1].

The sR-entropy of  $\Psi_{kAB}^A$  is  $n_3$  bits. After transmitting the information reconciliation value  $F' = F(\Psi_{kAB}^A)$  in step 6,  $A$  reveals at most an additional  $r' = n_3 h(\epsilon) + o(n)$  bits of sR-entropy.

<sup>6</sup>In the sequence-of-games technique, we make small changes to the security experiment, one after the other, beginning with the original security experiment. We must show that no adversary can distinguish any of the individual changes we made, and then that the final version of the experiment that we reach is secure.

Due to the laws of quantum mechanics, any attacker observing or modifying will alter  $A$ 's quantum transmission in proportion to the amount of information the attacker gains. In particular, we can obtain an upper bound on the amount of information learned by the attacker about  $\Psi_{kAB}^A$  based on the proportion of errors in the check bits  $\Psi_{chkAB}^A$  and  $\Psi_{chkAB}^B$ . If  $\epsilon$  is the proportion of errors, then the amount of information learned by any attacker is (except with probability exponentially small in  $n_3$ ) upper-bounded by  $2n_3h(\epsilon)$ .

Thus, the attacker's sR-entropy of  $\Psi_{kAB}^A$  conditioned on her attack is at least  $n_3(1 - 3h(\epsilon)) + o(n_3)$ . By applying privacy amplification in step 6(c) and 7(c), Theorem 4.7 of [CRE04] implies that the probability distribution on the final session key  $\Psi_{skAB}^A$  is  $\delta$  close to uniform, where  $\delta \leq 3 \cdot 2^{-\frac{n_3-r-s+1}{2}} + \text{negl}(k)$ , which is negligible in  $k$ . Thus no attacker can distinguish  $\Psi_{skAB}^A$  from a uniformly random string of the same length except with negligible probability. This argument shows that

$$\Pr(\text{Succ}_1) \leq \text{negl}(k) . \quad (2)$$

Combining equations (1) and (2), we obtain our result that  $|\Pr(\text{Succ}_0) - \frac{1}{2}| \leq \text{negl}(k)$ .  $\square$